Script – Présentation du projet AP4B

## Introduction

Bonjour et bienvenue dans cette présentation de notre projet d’AP4B : Le Munchkin.

## Présentation du projet

L’objectif du projet était de créer un programme ludique, développé en Java, dans lequel il serait possible de jouer à une version modifiée du célèbre jeu de société Munchkin.

Nous avons modifié les règles de bases pour simplifier le jeu, le rendant plus facile à être compris par des néophytes. Par exemple, il est impossible de tricher dans notre version du jeu. Nous avons aussi retiré la possibilité de placer plusieurs cartes lors d’un tour par le même joueur et les combats se passent en solo uniquement. La thématique du jeu est, quant à elle, adaptée à l’univers de l’UTBM.

Pour la partie technique du projet, nous avons choisi de séparer le coté fonctionnel du jeu et le coté interface graphique. Vous pouvez retrouver le code source dans les dossiers liées avec la vidéo. Cette séparation permet de simplifier le code et de le comprendre plus rapidement. La communication entre ces deux parties se fait via la classe App permettant la mise en place des règles du jeu.

Pour le fonctionnel, nous avons implémenté différents types de cartes : les classes, malédictions, langues, trésors, monstres et équipements. Le déroulement d’une partie nécessite des classes supplémentaires comme la pile de carte, les combats et les joueurs. Il y a aussi une classe obligatoire au bon fonctionnement du jeu : la classe CSVFileReader qui permet de charger la configuration depuis un fichier csv.

Pour l’interface graphique, nous avons utilisé la bibliothèque Swing car elle simple à utiliser et convient parfaitement pour la conception d’un programme à but non commercial ou à but démonstratif. Les classes s’organisent de la façon suivante : il y a la classe du menu principale, de la salle d’attente, du plateau de jeu et la classe de la fin de partie. Ces menus peuvent être vu comme des scènes que l’on affiche ou cache en fonction de la demande. Ces derniers fonctionnent grâce à des classes secondaires. On peut citer par exemple la classe CardButton qui gère l’affichage d’une carte sur le plateau.

## Fonctionnalités

Le jeu se déroule de la manière suivante : chaque joueur possède des cartes dans sa main. Au début du jeu, il en obtient 4. Le but du jeu est d’obtenir des niveaux par différents moyens et d’atteindre le niveau 10 pour gagner. Chacun peut tirer et jouer une carte par tour. Si la carte piochée est une carte monstre alors le joueur doit combattre le monstre ou le fuir. Pour gagner face au monstre, il faut que le joueur possède une puissance égale ou supérieure au monstre comme dans le jeu de société. Si le monstre est vaincu, le joueur a le droit de piocher un ou plusieurs cartes trésor et gagne des niveaux selon l’indication de la carte monstre. La puissance est égale à la somme du niveau du joueur et des points apportés par l’équipement. L’équipement peut être obtenu en piochant une carte lors de son tour ou en battant un monstre. Il y a 5 emplacements d’équipement par joueur correspondant aux parties du corps comme dans le jeu original. Il peut aussi piocher une carte malédiction, lui faisant perdre des niveaux. Si le joueur pioche un carte classe, langue ou équipement, il l’ajoute à sa main et pourra ensuite l’activer au tour actuel s’il n’a pas encore joué de carte ou à partir du tour suivant.

## Démo

<Parler ici en même temps qu’on enregistre un écran pour expliquer ce qu’on peut faire>

## Problèmes rencontrés et améliorations possibles

<Ici parler de problèmes rencontrés lors du développement>

Pour améliorer le programme, il aurait été possible d’ajouter un mode multijoueur en ligne pour améliorer l’expérience utilisateur et ne pas restreindre l’accès au jeu. Certains systèmes du jeu original ont été revus ou enlevés soit dû à leur difficulté d’implémentation soit dû à notre jugement quant à leur utilité

Problèmes :

- Comment recréer la multitude de cartes du jeu physique en ayant le moins de code possible

- Comment lier les effets des cartes au cartes en elles meme

- Jeu avec beaucoup de règles subtiles et difficile a implémenter

- Comment rendre l’interface intuitive, pratique et jolie

Améliorations possibles :

- Ajouter un mode en réseau

- Ajouter plus de fonctionnalités

- Ajouter des images pour les cartes

- Ajouter

## Conclusion

La démonstration est à présent terminée.

Merci pour votre attention